2022 CSP-S提高级第一轮认证 C++ 语言试题

**考生注意事项：**

● 试题纸共满分100 分。请在答题纸上作答，写在试题纸上的一律无效。

● 不得使用任何电子设备(如计算器、手机、电子词典等)或查阅任何书籍资料。

**一** **、** **单项选择题** **(共** **15题** **，每题2** **分** **，共计30** **分** **；** **每题有且仅有一个正确选项)**

1.在Linux系统终端中 ，用于切换工作日录的命令为 ()。

A.ls B.cd C.CP D.all

答案：B ls 命令用于显示指定工作目录下之内容（列出目前工作目录所含之文件及子目录)。 cd 命令用于切换当前工作目录。cp 命令主要用于复制文件或目录。all和Linux系统并无什么关 联。

2.你同时用time命令和秒表为某个程序在单核CPU的运行计时。假如time命令的输出如下： real 0m30.721s

user 0m24.579s

sys 0m6.123s

以 下最接近秒表计 时 的 时长为 ()。

A.30 s B.24 s C. 18 s D.6 s

答案：A Linux time 命令 用于统计给定命令所花费的总时间。它报告了以下几个关键

指标：实际时间 (Real Time)：从命令行开始执行到运行终止的实际消逝时间。

用户CPU时间(User CPU Time)：命令行执行完成时用户模式下CPU消耗的总时间。

系统CPU时间(System CPU Time)：命令行执行完成时内核模式下CPU消耗的总时间。 Real Time>User CPU Time+System CPU Time

3.若元素a、b、c、d、e、f依次进栈，允许进栈、退栈操作交替进行，但不允许连续三 次退栈操作 ， 则不可 能得到 的 出栈序列是 ()。

A.d ceb fa B.cbd a ef C.b cae fd D.a fed cb

答案 :D A选项abcd先依此进栈，d c再出栈，e进栈，e b出栈，f进栈后f a出栈。 BC 符合题目进栈退栈操作， D 连续退栈操作远超三次

4.考虑对n个数进行排序， 以下最坏时间复杂度低于O(n2)的排序方法是 ()。

A.插入排序 B.冒泡排序 C.归并排序 D.快速排序

答案:C 归并排序最环时间复杂度均为nlogn,插入排序和冒泡排序等简单排序方法的最坏时间复

杂度都是O(n2),而快速排序在最坏情况下则会退化为冒泡排序。

5. 假设在基数排序过程中，受宁宙射线的影响，某项数据异变为一个完全不同的值 。请问排

序算法结束后，可能出现的最坏情况是()。

A.移除受影响的数据后，最终序列是有序序列

B.移除受影响的数据后，最终序列是前后两个有序的子序列

C.移除受影响的数据后，最终序列是一个有序的子序列和 ·个基本无序的子序列

D.移除受影响的数据后，最终序列基本无序

【答案】A【解析】基数排序是按桶排序思想，某一个值的改变，不会造成其他数据排序失败， 排除这个数据，其余数据仍是有序序列

6. 计算机系统用小端( Little Endian)和大端( Big Endian)来描述多字节数据的存储地址顺序 模式，其中小端表示将低位字节数据存储在低地址的模式、大端表示将高位字节数据存储在低

地址的模式。在小端模式的系统和大端 模式的系统分别编译和运行以下C++代码段表示的程序， 将分别输出什么结果 ? ( )

unsigned x = 0xDEADBEEF;

unsigned char \*p = (unsigned char \*) & x;

printf("%X", \*p);

A.EF、EF B.EF、DE C.DE、EF D.DE、DE

【答案】B 低位是最低位的的EF，高位的就是最高位的DE

7.一个深度为5(根结点深度为1)的，按前序遍历的顺序给结点从1开始编号，则第100号结点的 父结点是第()号。

A.9 5 B.96 C.9 7 D.98

【答案】C 前序遍历的顺序是先根，再左子树，最后右子树。完全3叉树的第1层节点数1，

第2层3，第3层9，第4层27，第5层81，深度为5的完全3叉数节点总数为121个。根节点是1号 结点，深度为1。根结点的第一个儿子是2号结点，并且2号结点为根的子树一定是一棵深度 为4的满三叉树，因此有1+3+9+27=40个结点，所以根结点的第二个儿子是42号，根结点的三 号儿子是82号。 同理，可以得到82号的第一个儿子是83号，并且以83号为根的子树有

1+3+9=13个结点，因此82号的第二个儿子是96号，96号的第一个儿子是97号，而97号有三个 儿子分别是98号、99号和100号。

8.强连通图的性质不包括() :

A.每个顶点的度数至少为1 B.任意两个顶点之间都有边相连

C.任意两个顶点之间都有路径相连 D.每个顶点至少都连有一条边

【答案】 B 强连通图 ，任意两顶点都有路径可达即可 ，并不需要一定有边相连

9. 每个顶点度数均为2的无向图称为 “ 2正规图 ” 。 由编号为从1 到n的顶点 构成 的所 有 2 正规 图 ， 其 中包含 欧拉 回路 的 不 同 2正规 图 的数量为 ()。

A.n! B. (n-1) ! C.n!/2 D. (n-1) !/2

【答案】D 正规图理解为一个圆环排列，将某一个顶点固定进行排列则为An的全排列为(n- 1)!,那么欧拉回路顺时针与逆时针需要除以2去重，因此为 (n-1)!/2

10 . 共有8人选修了程序设计课程，期末大作业要求由2人组成的团队完成 。假设不区分 每个团队内2人的角色和作用，请问共有多少种可能的组队方案()。

A.28 B.32 C.56 D.64

【答案】A C=8\*7/2=28

11.小明希望选到形如“省A ·WDDD ”的车牌号。车牌号在“. ”之前的内容固定不 变； 后 面 的 5 位 号 码 中 ， 前 2 位 必 须 是 大 写 英 文 字 母 ， 后 3 位 必 须 是 阿 拉伯数字(里代表 A 至 Z,DD表 示 0 至 9 , 两 个 # 和 三 个DD之间可能相同也可 能 不 同) 。 请 问总共有 多少个可供 选择的车牌号 ()。

A.20280 B.52000 C.676000 D.1757600

【答案】C 大写字母26个，数字10个，26\*26\*10\*10\*10=676000

12.给定地址区间为0~9 的哈希表，哈希函数为 h(x)=x%10,采用线性探查的冲突解决策 略(对于出现冲突情况，会往后探查第一个空的地址存储； 若地址9冲突了则从地址0重 新开始探查) 。 哈希表初始为 空表 ，依次存储 ( 7 1 , 2 3 , 7 3 , 9 9 , 4 4 , 7 9 , 8 9 ) 后 ， 请 问 8 9 存储在哈希表哪个地址中。

A.9 B.0 C.1 D.2

【答案】D 存储情况见下表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 值 | 79 | 71 | 89 | 23 | 73 | 44 |  |  |  | 99 |

13.对于给定的n,分析以下代码段对应的时间复杂度，其中最为准确的时间复杂度为() 。 int i,j,k=0;

for(i=0;i<n;i++) {

for(j=1;j<n;j\*=2) {

k=k+n/2 ; }

}

A.0(n) B.0(nlogn) C.0(n τ) D.0(n²)

【答案】B 外循环次数n， 内循环次数因为是j=j\*2,所以循环次数为logn,时间复杂度为nlogn

14.以比较为基本运算，在n个数的数组中找最大的数，在最坏情况下至少要做 ( )次 运 算。

A.n/2 B.n-1 C.n D.n+1

【答案】B 最坏情况下需要每个数都比较一次，需要n-1次

15.ack函数在输入参数“ (2,2) ”时的返回值为()。 unsigned ack(unsigned m,unsigned n) {

if(m==0)return n+1 ;

if(n==0)return ack(m-1,1) ; return ack(m-1,ack(m,n-1)) ;

}

A.5 B.7 C.9 D.13

【答案】B 模拟计算一下 ack(1,0)=ack(0,1)=1+1=2

ack(1,1)=ack(0,ack(1,0))=ack(1,0)+1=3 ack(2,0)=ack(1,1)=3

ack(1,2)=ack(0,ack(1,1))=ack(0,3)=1+3=4 ack(1,3)=ack(0,ack(1,2))=ack(0,4)=1+4=5 ack(2,1)=ack(1,ack(2,0))=ack(1,3)=5

ack(1,4)=ack(0,ack(1,3))=ack(0,5)=1+5=6

ack(2,2)=ack(1,ack(2,1))=ack(1,5)=ack(0,ack(1,4))=1+6=7

二、阅读程序(程序输入不超过数组或字符串定义的范围；判断题正确填 √, 错误填× ;除 特殊说明外 ，判断题1.5分 ，选择题3分 ，共计40分)

(1)

1. #include<iostream>

2. #include<string>

3. #include<vector> 4.

5. usingnamespacestd; 6.

7. intf(conststring&s,conststring&t)

8. I{

9. int n=s.length(),m=t.length();

10.

11. vector<int>shift(128,m+1);

12.

13. int

i,j;

14.

15. for(j=0; j<m; j++)

16. shift[t[j]]=m-j;

17.

18. for(i=0; i<=n-m; i+=shift[s[i+m]]){

19. j=0;

20. while(j<m&&s[i+j]==t[j])j++;

21. if(j==m)return i;

22. }

23.

24. return -1;

25. }

26.

27. int main()

28.

{

30. c in >>a>>b;

32. return 0;

29. string a,b;

31. cout <<f(a,b)<<endl;

33. }

**假设输入字符串由ASCII可见字符组成，完成下面的判断题和单选题：**

● 判断题

16 . (1分)当输入为“abcde fg ” 时，输出为 - 1。 ( )

【答案】T 在s里查找t,abcde中没有fg,输出-1

17.当输入为 “abbababbbab abab”时 ， 输出为 4 。 ( )

【答案】F 第18行for(i=0; i<=n-m;i+=shift[s[i+m]]),循环从0开始 ，输出3。

18.当输入为 “GoodLuckCsp2022 22 ”时，第20行的 ”j++ ”语句执行次数为 2。( )

【答案】T 由于字符串t中只有字符‘2 ’,在s[i+m]!=‘2'时，i会直接加上m+1,而

m=2,因此i依次取值0、3、6、9、12,此时s[14]等于‘2 ’,所以下一个i是13,然后 s[13+0]==t[0],s[13+1]==t[1],最终得到答案13。只在i==13时执行过j++,共2次。

● 单选题

19.该算法最坏情况下的时间复杂度为( )。

A.0(n+m) B.0(nlogm) C.0(mlogn) D.O(nm)

【答案】D 形如s=aaa....aaaa t=aaab时，字符a跳跃的最小值为2，复杂度为 O(nm)

20.f(a,b)与下列()语句的功能最类似。

A.a.find(b)

C.a.substr(b)

【答案】A 在字符串a中找字符串b第一次出现的位置。

B.a.rfind(b)

D.a.compare(b)

21.当输入为 “baaabaaabaaabaaaa aaaa ”,第20行的 “j++ ”语句执行次数为 ():

A.9 B.10 C.11 D.12

【答案】B 模拟时要注意会跳过一些a。i的取值分别是0、5、6、7、8、13,其中 i=5时，j++ 运行3次；i=6时，j++运行2次；i=7时，j++运行1次；i=13时，j++运行4次，共10次。

(2)

1. #include<iostream>

2.

3. using namespace std;

4.

5. const int MAXN=105;

6.

7. int n,m,k,val[MAXN];

8. int temp[MAXN],cnt[MAXN];

9.

10. void in it()

11. {

12. c in >>n>>k;

13. for(inti=0;i<n;i++)c in>>val[i];

14. int maximum=val[];

15. for(inti=1;i<n;i++)

16. if(val[i]>maximum)maximum=val[i];

17. m=1;

18. while(maximum >=k){

19. maximum/=k;

20. m++;

21. } 22. }

23.

24. void solve()

25. {

26. int base =1;

27. for(inti=0;i<m;i++){

28. for(int j=0;j<k;j++)cnt[j]=0;

29. for(intj=0;j<n;j++)cnt[val[j]/base%k]++;

30. for(int j=1;j<k;j++)cnt[j]+=cnt[j-1];

31. for(intj=n-1;j>=0;j--){

32. temp[cnt[val[j]/base %k]-1]=val[j];

33. cnt[val[j]/base%k]--;

34. }

35. for(intj=0;j<n;j++)val[j]=temp[j];

36. base \*=k;

37. } 38. }

39.

40. int main()

41. {

42. in it();

43. solve();

44. for(int i=0;i<n;i++)cout<<val[i]<< ' ';

I 45. cout <<endl;

46. return 0;

47. }

**假设输入的n为不大于100的正整数，k为不小于2且不大于100的正整数,val[i]在** **int表示范围内，完成下面的判断题和单选题：**

**此代码是k进制基数排序**。

● 判断题

22.这是一个不稳定 的排序算法 。 ( )

【答案】F 基数排序是稳定排序

23.该算法的空间复杂度仅与n有关。 ( )

【答案】F 数组cnt的大小与n和k都有关，空间复杂度是0(n+k)

24.该算法的时间复杂度为0(m(n+k)) 。 ( )

【答案】T 外循环m， 内循环(n+k)

● 单选题

25. 当输入为 “ 5 3 98 26 91 37 46 ”时 ，程序第一次执行到第36行 ，val[ ]数 组 的 内 容 依 次 为 ( )。

A.91 26 46 37 98 B.91 46 37 26 98

C.98 26 46 91 37 D.91 37 46 98 26

【答案】D 3进制下的第一位数的比较,91,37,46,98 ,26 的三进制 最后一位 是 1, 1, 1 ,2,2。

26.若val[i]的最大值为100,k取()时算法运算次数最少。

A.2 B.3 C.10 D. 不确定

【答案】D 运算次数不仅和k有关，还跟n有关

27.当输入的k比val[i]的最大值还大时，该算法退化为( )算法。

A. 选择排序 B. 冒泡排序 C. 计数排序 D. 桶排序

【答案】C k>max(val[i])时，算法成为了计数排序。

(3)

1. #include<iostream>

2. #include<algorithm>

3.

4. using namespace std;

5.

6. const int MAXL=1000;

7.

8. int n,k,ans[MAXL];

9.

10. int main(void)

{

11.

12. c in >>n>>k;

13. if(!n)cout<<0<<endl;

14. else

15. {

16. int m =0;

17. while(n)

18. {

19. ans[m++]=(n%(-k)+k)%k;

20. n=(ans[m-1]-n)/k;

21. }

22. for(inti=m-1;i>=0;i--)

23. cout<<char(ans[i]>=10?

24. ans[i]+ 'A '-10:

25. ans[i]+ '0 ');

26. cout<<endl;

27. }

28. return 0;

29. } 30.

**假设输入的n在int范围内，k为不小于2且不大于36** **的正整数，完成下面的判断题和单** **选题：**

● 判断题

28.算法的时间复杂度为O(logkn)。 ()

【答案】T 每次循环后n除以k

29.删除第23行的强制类型转换，程序的行为不变 。 ()

【答案】F 不强制类型转换输出的是int类型的值

30.除非输入的 n为0,否则程序输出的字符数为O(L logk |n|」+1)()

【答案】T 有余数时+1

● 单选题

31.当输入为“ 100 7 ”时，输出为()。

A.202 B.1515 C.244 D.1754

【答案】A 31-33题可以直接模拟计算一下

32.当输入为“-255 8 ”时，输出为“ () ”。

A.1400 B.1401 C.417 D.400

【答案】B

33.当输入为“1000000 19”时，输出为“ ()”。

A.BG939 B.87GIB C.1CD428 D.7CF1B

【答案】B

**三、完善程序(单选题，每小题3** **分，共计30** **分)**

**(1)(归并第k小** **)** 已知两个长度均为n的有序数组al和a2(均为递增序，但不保证严格单调递 增),并且给定正整数 k(1≤k≤2n),求数组al和a2归并排序后的数组里第k小的数值。

试补全程序。

1. #include<bits/stdc++.h>

2. using namespace std;

3.

4. int solve(int \*al,int \*a2,int n,int k){

5. intleft1=0,right l=n-1;

6. int left2=0,right2=n-1;

7. while(left1<=right1&&left2<=right2){

8. int ml=(left l+right l)>>1;

|  |  |
| --- | --- |
| 9. int m2=(left2+right2)>>1; | |
| 10. int cnt =①; | |
| 11. | if(② ){ |
| 12. | if(cnt<k)left1=ml+1; |
| 13. | elseright2=m2-1; |
| 14. | }else { |
| 15. | if(cnt<k)left2=m2+1; |
| 16. | else right1=ml-1; |
| 17. | } |
| 18. | } |
| 19. | if(③ ){ |
| 20. | if(left1==0){ |
| 21. | return a2[k-1]; |
| 22. | }else { |
| 23. | intx=al[left1-1],④ ; |
| 24. | return std::max(x,y) |
| 25. | } |
| 26. | }else{ |
| 27. | if(left2==0){ |
| 28. | return al[k-1]; |
| 29. | }else { |
| 30. | int x=a2[left2-1],⑤; |
| 31. | returnstd::max(x,y); |
| 32. | } |
| 33. | } |
| 34. } | |
| 35. | |

34.① 处 应 填 ( )

A. (ml+m2)\*2 B. (ml-1)+(m2-1) C.ml+m2 D. (m1+1)+(m2+1)

【答案】C 用 来 记 录 合 并 数 组 当 前 位 置 的 下 标

35.② 处 应 填 ( )

A.a1[ml]==a2[m2] B.a1[ml]<=a2[m2] C.a1[ml]>=a2[m2] D.a1[m1]!=a2[m2]

【答案】B 比 较 两 个 数 组 对 应 的 中 间 位 置 的 大 小

36. ③ 处 应 填 ( )

A.left1==right1 B.left1<right1 C.left1>right1 D.left1!=right1

【答案】C 二分结束条件是 left1>right1或者left2>right2

37.④ 处 应 填 ( )

A.y=a1[k-left2-1] B.y=a1[k-left2] C.y=a2[k-left1-1] D.y=a2[k-left1]

【答案】C a2数组组成 k个数的数量

38. ⑤ 处 应 填 ( )

A.y=a1[k-left2-1] B.y=a1[k-left2] C.y=a2[k-left1-1] D.y =a2[k-left1]

【答案】A a1数组组成k个数的数量

**(2)(容器分水)** 有两个容器 ，容器1的容量为为a升 ，容器2的容量为b升； 同时允许下 列的三种操作，分别为：

1)FILL(i):用水龙头将容器i(i∈ {1,2})灌满水； 2)DROP(i):将容器i的水倒进下水道；

3)POUR(i,j):将容器i的水倒进容器j(完成此操作后，要么容器j被灌满 ，要么容器 i 被清空)。

求只使用上述的两个容器和三种操作，获得恰好c升水的最少操作数和操作序列。 上述a 、b 、c均为不超过100的正整数，c≤max{a,b}。

试补全程序。

1. #include<bits/stdc++.h>

2. using namespace std;

3. constintN=110; 4.

5. intf[N][N]; 6. int ans;

7. inta,b,c; 8. int in it;

9.

10. int dfs(int x,int y){

11. if(f[x][y] !=in it)

12. return f[x][y];

13. if(x==c ||y==c)

14. return f[x][y]=0;

15. f[x][y]=in it-1;

16. f[x][y]=min(f[x][y],dfs(a,y)+1);

17. f[x][y]=min(r[x][y],dfs(x,b)+1);

18. f[x][y]=min(f[x][y],dfs(0,y)+1);

19. f[x][y]=min(f[x][y],dfs(x,0)+1);

20. int t=min(a-x,y);

21. f[x][y]=min(f[x][y],① );

22. t=min(x,b-y);

23. f[x][y]=min(f[x][y],② );

24. return f[x][y];

25. } 26.

27. voidgo(intx,inty){

28. if(③ )

29. return;

30. if(f[x][y]==dfs(a,y)+1){

31. cout<<"FILL(1)"<<endl;

32. go(a,y);

33. }elseif(f[x][y]==dfs(x,b)+1){

34. cout <<"FILL(2)"<<endl;

35. go(x,b);

36. }else if(f[x][y]==dfs(0,y)+1){

37. cout<<"DROP(1)"<<endl;

38. go(0,y);

39. }elseif(f[x][y]==dfs(x,0)+1){

40. cout<<"DROP(2)"<<endl;

41. go(x,0);

42. }else {

43. intt=min(a-x,y);

44. if(f[x][y]==④){

45. cout<<"POUR(2,1)"<<endl;

46. go(x+t,y-t);

47. }else{

48. t=min(x,b-y);

49. if(f[x][y]==⑤){

50. cout<<"POUR(1,2)"<<endl;

51. go(x-t,y+t);

52. }else

53. assert(0);

54. }

55. } 56. }

57.

58. int main(){

59. c in>>a>>b>>c;

60. ans=1<<30;

61. memset(r,127,sizeoff);

62. in it=\*\*f;

63. if((ans=dfs(0,0))==in it-1)

64. cout<<"impossible";

65. else{

66. cout<<ans<<endl;

67. go(0,0);

68. }

69. }

39. ① 处应填 ()

A.dfs(x+t,y-t)+1 B.dfs(x+t,y-t)- 1 C.dfs(x-t,y+t)+1 D.dfs(x-t,y+t)- 1

【答案】A 将容器2中的水倒入容器1中

40 . ②处应填 ()

A.dfs(x+t,y-t)+1 B.dfs(x+t,y-t)- 1 C.dfs(x-t,y+t)+1 D.dfs(x-t,y+t)- 1

【答案】C 将容器1中的水倒入容器2中，和39题是对称的

41. ③ 处应填 ()

A.x==c||y==c B.x==c&&y==c C.x>=c||y>=c D.x>=c&&y>=c

【答案】A 判断是否已经找到 C 升的水

42. ④ 处应填 ()

A.dfs(x+t,y-t)+1 B.dfs(x+t,y-t)-1 C.dfs(x-t,y+t)+1 D.dfs(x-t,y+t)-1

【答案】A判断下一步是否为将容器2中的水倒入容器1中

43 . ⑤处应填 ()

A.dfs(x+t,y-t)+1 B.dfs(x+t,y-t)-1 C.dfs(x-t,y+t)+1 D.dfs(x-t,y+t)-1

【答案】C 判断下一步是否为将容器1中的水倒入容器2中，和42题是对称的

**2022年CSP-S1提高组第一轮参考答案**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **答案** |  | **序号** | **答案** |
| 1 | B |  | 9 | D |
| 2 | A |  | 10 | A |
| 3 | D |  | 11 | C |
| 4 | C |  | 12 | D |
| 5 | A |  | 13 | B |
| 6 | B |  | 14 | B |
| 7 | C |  | 15 | B |
| 8 | B |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **答案** |  | **序号** | **答案** |
| 16 | √ |  | 22 | X |
| 17 | × |  | 23 | × |
| 18 | √ |  | 24 | √ |
| 19 | D |  | 25 | D |
| 20 | A |  | 26 | D |
| 21 | B |  | 27 | C |

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **答案** | **序号** | **答案** |
| 28 | √ | 34 | C |
| 29 | X | 35 | B |
| 30 | √ | 36 | C |
| 31 | A | 37 | C |
| 32 | B | 38 | A |
| 33 | B |  |  |

|  |  |
| --- | --- |
| **序号** | **答案** |
| 39 | A |
| 40 | C |
| 41 | A |
| 42 | A |
| 43 | C |